UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/071,526	02/08/2002	Jonathan A. Forbes	3382-61916	2616

26119 7590 08/24/2006

KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER	
RAMPURIA, SATISH	
ART UNIT	**PAPER NUMBER**
2191	

DATE MAILED: 08/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/071,526	FORBES ET AL.	
	Examiner	Art Unit	
	Satish S. Rampuria	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 May 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15,17-20 and 22-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15,17-20 and 22-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *5/25/06*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Response to Amendment

1. This action is in response to the amendment received on May 25/2006.

2. Claim previously cancelled by the Applicant: 16, 21.

3. Claims currently amended by the Applicant: 1, 12, 13, 17, 24, 26, 27 and 28.

4. Claims pending in the application: 1-15, 17-20 and 22-32.

Information Disclosure Statement

5. An initialed and dated copy of Applicant's IDS form 1449 filed on May 25/2006 is attached to the instant Office action. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Specification

6. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: A method and system for processing software dependencies in management of software package.

Response to Arguments

7. Applicant's arguments with respect to claims have been considered but are moot in view of new ground(s) of rejection.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-9, 11-15, 17-20, 22-29 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,586,304 to Stupek, Jr. et al. (hereinafter called Stupek) in view of US Patent No. 5,493,682 to Tyra et al., (hereinafter called Tyra) and further in view of US Patent No. 6,349,408 to Smith.

Per claims 1:

Stupek disclose:

- determining whether software associated with the software dependency is present on the computer (col. 6, lines 24-25 "The database also contains information regarding the dependencies"); and
- responsive to determining the software associated with the software dependency is not present on the computer, acquiring the software associated with the software dependency (col. 6, lines 45-48 "dependency information in the Package database 25 describes not only the dependencies between packages on the CD, but also all dependencies between an upgrade package and any upgrade not available on the CD");
- after acquiring the software associated with the software dependency (col. 7, lines 50-53 "collecting information about the corresponding package... pointers to parent, child and sibling packages"), updating a database at the computer indicating the software

associated with the software dependency is installed on the computer (col. 7, lines 5-10 "Within the How_To database, each record represents an individual piece of MIB information corresponding to the particular package... upgrade device... specified in the record");

- wherein at least one of the software dependencies refers to a list comprising one or more other software dependencies (col. 6, lines 24-31 "child dependencies 25h are the upgrade objects associated with a package; sibling dependencies 25j are the packages upon which a package depends; and parent dependencies").

Stupek does not explicitly disclose wherein the acquiring the software associated with the software dependency comprises recursively processing the one or more other software dependencies.

However, Tyra discloses in an analogous computer system wherein the acquiring the software associated with the software dependency comprises recursively processing the one or more other software dependencies (col. 5, lines 19-30 "...object dependency interpreter is a recursive process that evaluates each software object to determine its dependencies, and ensures that all lower level dependency conditions have been satisfied before the present object is evaluated").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of recursive processing the software dependencies as taught by Tyra into the method of automatic upgrading the program/software as taught by Stupek. The modification would be obvious because of one of ordinary skill in the

art would be motivated to use the recursive process to maintain the large software systems comprisd of a large number of software components as suggested by Tyra (col. 3, lines 36-40).

Neither Stupek nor Tyra explicitly disclose *acquiring, by a software package manager running on the computer*, the software associated with the software dependency.

However, Smith discloses in an analogous computer system *acquiring, by a software package manager running on the computer*, the software associated with the software dependency (FIG. 6 and FIG. 10; col. 6, lines 4-21 and lines 45-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of *acquiring, by a software package manager running on the computer*, the software associated with the software dependency as taught by Smith into the method of automatic upgrading the program/software as taught by the combination system of Stupek and Tyra. The modification would be obvious because of one of ordinary skill in the art would be motivated to acquire the software package manager running on the computer to have automatic techniques for installing an application for extending the features of an application that been installed as suggested by Smith (col. 1, lines 50-62).

Per claim 2:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- wherein acquiring the software associated with the software dependency comprises acquiring a file comprising the list comprising one or more other software dependencies

(col. 6, lines 24-26 "The database also contains information regarding the dependencies between the package and other upgrade objects or packages").

Per claim 3:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- wherein acquiring the software associated with the software dependency comprises acquiring a list of one or more files from a remote location and acquiring the files in the list (col. 3, lines 56-58 "the upgrade device 10 automatically analyzes each network resource 3 currently on the server 1 to determine the availability and necessity of the corresponding upgrade 7").

Per claim 4:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- wherein one or more of the software dependencies is associated with a location whereat the list of other software dependencies can be found. The limitation of this claims are similar to those in claim 3 and rejected under the same rational.

Per claims 5 and 6:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- wherein the database is operable to indicate whether a plurality of software components are installed via a single name associated with the plurality of software components (col.

6, lines 24-26 "The database also contains information regarding the dependencies between the package and other upgrade objects or packages").

Per claim 7:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- one or more dependencies in the list of software dependencies is associated with a version number (col. 4, lines 24-28 "upgrade advisor places information about the resource (e.g., name, version number) into a driver table...the server manager located in the server uses this information to search for the resource (i.e., to see if the resource has been installed on the network"); and
- determining the dependency is not present on the computer comprises determining software satisfying the version number is not present on the computer (col. 1, lines 60-62 "storing upgrade information which identifies the later version and describes features of the later version relative to one or more earlier version").

Per claim 8:

The rejection of claim 7 is incorporated, and further, Stupek disclose:

- wherein at least two software dependencies are associated with different version numbers. The limitation of this claims are similar to those in claim 7 and rejected under the same rational.

Per claim 9:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- wherein at least one of the dependencies specifies a plurality of software items forming a software package (col. 6, lines 24-31 "child dependencies 25h are the upgrade objects associated with a package; sibling dependencies 25j are the packages upon which a package depends; and parent dependencies").

Per claim 11:

The rejection of claim 11 is incorporated, and further, Stupek disclose:

- wherein acquiring dependencies is deferred until execution of software associated with the dependencies is requested (col. 6, lines 24-31 "child dependencies 25h are the upgrade objects associated with a package; sibling dependencies 25j are the packages upon which a package depends; and parent dependencies" and col. 6, lines 24-26 "The database also contains information regarding the dependencies between the package and other upgrade objects or packages").

Claim 12 is the computer program product claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.

Per claims 13 and 14:

Stupek disclose:

- specifying a name of the software dependency (col. 6, line 24-25 "database contains information regarding the dependencies"), wherein the name is operable to identify a list

of one or more other software dependencies (col. 6, lines 24-31 "child dependencies 25h are the upgrade objects associated with a package; sibling dependencies 25j are the packages upon which a package depends; and parent dependencies");

- specifying a version of the software dependency (col. 2, lines 46-47 "information regarding dependencies between upgrades is provided");
- comparing the version for the software dependency against a version of software installed at the computer(col. 4, lines 5-7 "The upgrade advisor 11 then retrieves upgrade information from the upgrade database 9 and performs two types of comparisons"); and
- responsive to determining the version installed at the computer is not sufficient, installing the software dependency (col. 3, lines 56-58 "upgrade device 10 automatically analyzes each network resource 3 currently on the server 1 to determine the availability and necessity of the corresponding upgrade 7").

Stupek does not explicitly disclose wherein the acquiring the software associated with the software dependency comprises recursively processing the one or more other software dependencies.

However, Tyra discloses in an analogous computer system wherein the acquiring the software associated with the software dependency comprises recursively processing the one or more other software dependencies (col. 5, lines 19-30 "...object dependency interpreter is a recursive process that evaluates each software object to determine its dependencies, and ensures that all lower level dependency conditions have been satisfied before the present object is evaluated").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of recursive processing the software dependencies as taught by Tyra into the method of automatic upgrading the program/software as taught by Stupek. The modification would be obvious because of one of ordinary skill in the art would be motivated to use the recursive process to maintain the large software systems comprisd of a large number of software components as suggested by Tyra (col. 3, lines 36-40).

Neither Stupek nor Tyra explicitly disclose *acquiring, by a software package manager running on the computer*, the software associated with the software dependency.

However, Smith discloses in an analogous computer system *acquiring, by a software package manager running on the computer*, the software associated with the software dependency (FIG. 6 and FIG. 10; col. 6, lines 4-21 and lines 45-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of *acquiring, by a software package manager running on the computer*, the software associated with the software dependency as taught by Smith into the method of automatic upgrading the program/software as taught by the combination system of Stupek and Tyra. The modification would be obvious because of one of ordinary skill in the art would be motivated to acquire the software package manager running on the computer to have automatic techniques for installing an application for extending the features of an application that been installed as suggested by Smith (col. 1, lines 50-62).

Per claim 15:

The rejection of claim 13 is incorporated, and further, Stupek disclose:

- wherein the software dependency is associated with a software package depending on at least one other software package (col. 6, lines 24-26 "database... contains information... dependencies between the package and other upgrade objects or packages").

Per claim 16: (Previously Cancelled)

Per claims 17 and 23:

Stupek disclose:

- consulting a database to see if software associated with the name is already installed at the computer (col. 4, lines 7-12 "a) whether or not a particular upgrade package corresponds to a resource on the server, and b) whether or not the version number of the upgrade package matches the version number of the corresponding network resource (i.e., whether or not the upgrade package represents a true upgrade for the existing network resource)"); and
- responsive to determining software associated with the name is not already installed at the computer, acquiring the specified software (col. 3, lines 56-58 "upgrade device 10 automatically analyzes each network resource 3 currently on the server 1 to determine the availability and necessity of the corresponding upgrade 7"); and
- responsive to determining software dependencies associated with the specified software are not already installed at the computer, acquiring the software dependencies (col. 3, lines 56-58 "upgrade device 10 automatically analyzes each network resource 3 currently

on the server 1 to determine the availability and necessity of the corresponding upgrade 7");

- wherein the name is operable to specify a plurality of software components (col. 6, lines 24-31 "child dependencies 25h are the upgrade objects associated with a package; sibling dependencies 25j are the packages upon which a package depends; and parent dependencies").

Stupek does not explicitly disclose wherein acquiring the specified software* comprises recursively processing software dependencies associated with the name to find one or more other software dependencies associated with names designating software.

However, Tyra discloses in an analogous computer system wherein acquiring the specified software* comprises recursively processing software dependencies associated with the name to find one or more other software dependencies associated with names designating software (col. 5, lines 19-30 "...object dependency interpreter is a recursive process that evaluates each software object to determine its dependencies, and ensures that all lower level dependency conditions have been satisfied before the present object is evaluated").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of recursive processing the software dependencies as taught by Tyra into the method of automatic upgrading the program/software as taught by Stupek. The modification would be obvious because of one of ordinary skill in the art would be motivated to use the recursive process to maintain the large software systems comprisd of a large number of software components as suggested by Tyra (col. 3, lines 36-40).

Neither Stupek nor Tyra explicitly disclose *acquiring, by a software package manager running on the computer*, the software associated with the software dependency.

However, Smith discloses in an analogous computer system *acquiring, by a software package manager running on the computer*, the software associated with the software dependency (FIG. 6 and FIG. 10; col. 6, lines 4-21 and lines 45-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of *acquiring, by a software package manager running on the computer*, the software associated with the software dependency as taught by Smith into the method of automatic upgrading the program/software as taught by the combination system of Stupek and Tyra. The modification would be obvious because of one of ordinary skill in the art would be motivated to acquire the software package manager running on the computer to have automatic techniques for installing an application for extending the features of an application that been installed as suggested by Smith (col. 1, lines 50-62).

Per claim 18:

The rejection of claim 17 is incorporated, and further, Stupek disclose:

- wherein the plurality of software components is divided among a plurality of files (col. 5, lines 30-37).

Per claim 19:

The rejection of claim 17 is incorporated, and further, Stupek disclose:

- wherein the plurality of software components comprises a plurality of files. The limitation of this claims are similar to those in claim 18 and rejected under the same rational.

Per claim 20:

- wherein the name is operable to specify a plurality software components; and different versions are associated with the different software components (col. 4, lines 7-12 "a) whether or not a particular upgrade package corresponds to a resource on the server, and b) whether or not the version number of the upgrade package matches the version number of the corresponding network resource (i.e., whether or not the upgrade package represents a true upgrade for the existing network resource)").

Per claim 21: (Cancelled)

Per claim 22:

The rejection of claim 17 is incorporated, and further, Stupek disclose:

- for at least on software dependencies, determining software associated with the dependency is already installed at the computer(col. 3, lines 56-58 "upgrade device 10 automatically analyzes each network resource 3 currently on the server 1 to determine the availability and necessity of the corresponding upgrade 7"), wherein the dependency specifies a plurality of software components (col. 3, lines 56-58 "upgrade device 10 automatically analyzes each network resource 3 currently on the server 1 to determine the availability and necessity of the corresponding upgrade 7").

Per claim 32:

The rejection of claim 1 is incorporated, and further, Stupek disclose:

- after acquiring the software associated with the software dependency, installing the software (col. 3, lines 64-67 "To determine which upgrades 7 should be installed to the server, the upgrade advisor 11 retrieves information about the MIB 5 from a server database 13 located in the server manager").

Claims 24 and 26 are the computer program product claim corresponding to method claim 17 and rejected under the same rational set forth in connection with the rejection of claim 17 above.
Claim 25 is the computer program product claim corresponding to method claim 20 and rejected under the same rational set forth in connection with the rejection of claim 20 above.
Claim 27 is the computer program product claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.
Claim 28 is the system claim corresponding to method claim 17 and rejected under the same rational set forth in connection with the rejection of claim 17 above.
Claim 29 is the system claim corresponding to method claim 26 and rejected under the same rational set forth in connection with the rejection of claim 26 above.

10. Claims 10, 30, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stupek, Tyra, Smith and further in view of US Patent No. 6,802,061 to Parthasarathy (hereinafter called Parthasarathy) .

Per claim 10:

Stupek does not explicitly disclose software package comprises a mixture of native code components and Java classes.

However, Parthasarathy discloses in an analogous computer system software package comprises a mixture of native code components and Java classes (col. 3, lines 18-24 "any software component can be downloaded, verified, and installed... whether it is JAVA class library...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of downloading any software components as taught by Parthasarathy into the method of upgrading the computer as taught by the combination of Stupek, Tyra and Smith. The modification would be obvious because of one of ordinary skill in the art would be motivated to have the mixture of software package components to provide versatile types of updates as suggested by Parthasarathy (col. 3, lines 42-49).

Per claim 30:

Stupek does not explicitly disclose wherein the other list of dependencies is specified via an URL.

However, Parthasarathy discloses in an analogous computer system wherein the other list of dependencies is specified via an URL (col. 3, lines 14-6 "software component download module for locating computer software components with uniform resource locators (URLs)").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of downloading any software components using URLs as taught by Parthasarathy into the method of upgrading the computer as taught by the combination of Stupek, Tyra and Smith. The modification would be obvious because of one of ordinary skill in the art would be motivated to have the URLs of software package components to provide versatile types of updates as suggested by Parthasarathy (col. 3, lines 42-49).

Per claim 31:

Stupek does not explicitly disclose comprising a browser; wherein the software package manager is operable to initiate execution of a software package as directed by the browser upon encountering HTML tags indicating the specified list of dependencies.

However, Parthasarathy discloses in an analogous computer system comprising a browser; wherein the software package manager is operable to initiate execution of a software package as directed by the browser upon encountering HTML tags indicating the specified list of dependencies (col. 3, lines 34-38 "When the <OBJECT> tag is encountered in a HTML document during browsing with a network browser, the multimedia software components referenced by the <OBJECT> tag are automatically downloaded and displayed directly on user's computer").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of downloading any software components using HTML tags as taught by Parthasarathy into the method of upgrading the computer as

taught by the combination of Stupek, Tyra and Smith. The modification would be obvious because of one of ordinary skill in the art would be motivated to have the HTML tags of software package components to provide versatile types of updates as suggested by Parthasarathy (col. 3, lines 42-49).

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571) 272-3732**. The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to Friday except every other Friday and federal holidays. Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: 571-272-2100**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Wei Y. Zhen** can be reached on **(571) 272-3708**. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner/Software Engineer
Art Unit 2191
8/21/06

WEI ZHEN
SUPERVISORY PATENT EXAMINER